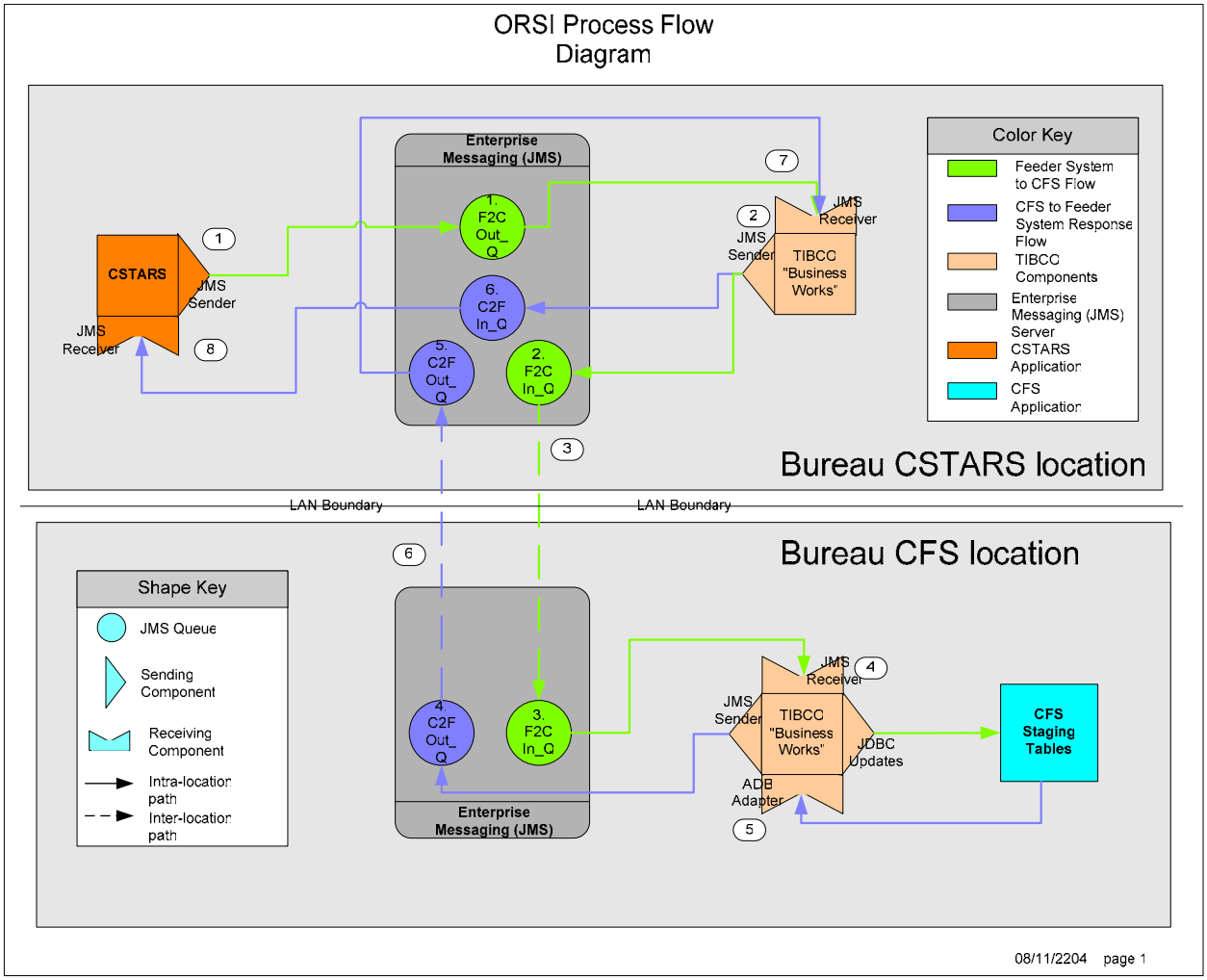
TIBCO Data Flow

1 Overview

The figure below shows how data will flow in TIBCO between CSTARS and CFS. The green arrows and objects show how data will flow from CSTARS to CFS (Section 1.1 below). The blue arrows and objects show how data will flow back from CFS to CSTARS (Section 1.2 below). The numbers on the chart correspond to detailed process descriptions listed in Section 2.



1.1 Feeder System to CFS Conceptual Design

Messages flow from the Feeder system to TIBCO for processing. To accomplish this, CSTARS renders an XML message containing the necessary data and send it to TIBCO via JMS queues. The message is received and parsed by TIBCO and routed to the appropriate JMS queue. The message is then forwarded, if needed, to the

corresponding JMS server of the receiving application. JMS consumers within the TIBCO business processes pull the message from the JMS server and perform the required data translation. After the data has been translated, it is inserted into the CFS staging tables. The diagram below repeats the conceptual flow diagram but highlights the message path for Feeder to CFS flow in green.

1.2 CFS to Feeder System Conceptual Design

After executing CFS validation processes, the transaction response data is inserted into the CFS staging tables (out-queues). The TIBCO Active Database subscriber services pick up the response and bring it into the TIBCO Business Works engine. The message is then routed to the appropriate JMS queue name based on the source and destination systems. If necessary, the JMS server forwards the message to the JMS server of the corresponding receiving system. NIST is the exception to this since its CSTARS and CFS applications reside within the same location. JMS consumers within the TIBCO business processes pull the message from the JMS server and perform the required data translation. After the data has been translated, it is rendered into an XML message and sent to the appropriate JMS queue where it is received by the CSTARS JMS consumer and inserted into the CSTARS database tables. The blue in the diagram below highlights the flow of the messages from CFS to the Feeder system.

2 Process Flow

1. CSTARS renders an XML message containing the transaction data and inserts it into the body of a JMS message. JMS senders configured on the CSTARS system send this message to a queue on the JMS server of the sending location. This queue grouping is called the F2C_Out_Q. This grouping contains all messages sent from the feeder system to CFS before being processed by TIBCO. In this grouping there are separate queues for Obligations and Requisitions. The CSTARS sending application sends the message to the appropriate queue depending on the transaction type. Please refer to the JMS queue configuration diagram (Figure 5) for further details.
2. Processes configured within TIBCO BusinessWorks receive JMS messages from the F2C_Out_Q grouping on the JMS server of the sending application location. One process receives Obligation messages and another receives Requisitions. A message sent to the Obligation queue or the Requisition queue from CSTARS initiates the appropriate process. The screenshot below shows the TIBCO BusinessWorks processes that receives the JMS message, parses and maps the data and sends the reconfigured XML to the appropriate JMS queue in the F2C_In_Q grouping to execute the necessary routing. As mentioned above, separate instances of this process exist for Requisitions and Obligations. Each performs the same respective functionality but has different JMS receivers and XML schema definitions (XSD) for Requisitions and Obligations, respectively. Sections 2.1 – 2.9 describe these TIBCO activities in more detail. The diagram below contains the Level 1 distinction because it is a main process. A sub-process is called at step 2.6 which transforms the format sent from CSTARS to the shared data object (SDO). This sub-process is listed as a Level 2 process in Figure 8.

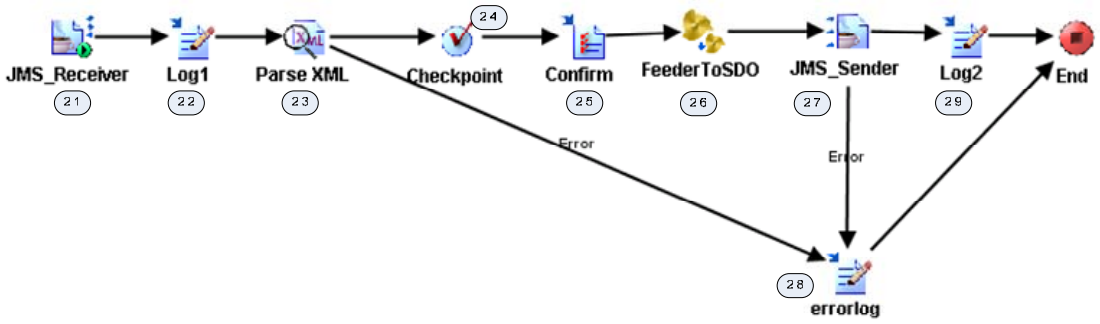


Figure 1: Feeder to CFS Outbound Process – Level 1

2.1 A configured JMS receiver listens to the F2C_Out_Q grouping. The JMS receiver for Obligations is configured to receive messages with the queue name of <env>.f2coutq.cstars.cfs.obligation. The JMS receivers for Requisitions are configured to receive messages with the queue name of <env>.f2coutq.cstars.cfs.requisitions.

2.2 The next activity writes a custom message to a log file stored within the TIBCO BusinessWorks installation directory under the \logs subdirectory. This message can be viewed through the TIBCO Administration GUI. This custom message is tailored to provide useful information which the system administrator uses for tracking and debugging purposes. This log file records the queue from which the message came, a time stamp, and the unique identifier of the message.

2.3 The XML string within the body of the JMS message is parsed using a Parse XML activity and the appropriate XSD. One XSD is created for Requisitions and one for Obligations since each contains different fields. Each is used in its respective process. This activity separates the raw XML string into name/value pairs for translation, interpretation, and mapping as needed.

If an error occurs during the XML parsing activity, the process transitions to the error log activity and capture the necessary error information (see step 2.7 for additional information). For example, if improperly formed XML is passed, the activity is unable to parse the data correctly and an error occurs.

2.4 After the TIBCO BusinessWorks engine receives the JMS message, it executes a checkpoint activity. A checkpoint saves the current process data and state so that it can be recovered at a later time in the event of a failure. If a process engine fails, all process instances can be recovered and resume execution at the location of their last checkpoint in the process definition.

2.5 The Confirm activity confirms any confirmable messages received by the process instance. For ORSI purposes, it confirms that BusinessWorks process received the JMS message. When this occurs, the JMS server removes the message from the queue.

2.6 This activity calls the sub-process that transforms the dataset sent by CSTARS to the shared data object (SDO) format. This functionality is in a sub-process so that it may be called by other processes if the need arises. The SDO also decouples the CSTARS and CFS formats. If other entities want to leverage the enterprise integration framework in

the future, only the transformation to/from the SDO needs to be developed. The diagram and steps below describe the activities within this sub-process.

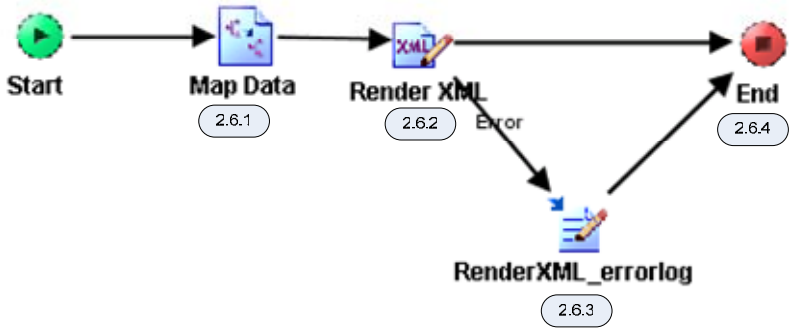


Figure 2: Feeder to SDO Transformation sub-process – Level 2

2.6.1 This Map Data activity transforms information sent from the CSTARS schema to the SDO schema. See Appendix A for details regarding this mapping.

2.6.2 The output of the Map Data activity is rendered into an XML message which is sent in the body of the JMS message to the appropriate JMS queue. If an error occurs while rendering the XML, the process transitions to the RenderXML_errorlog. See the step 2.6.3 for further information.

2.6.3 The RenderXML_errorlog creates a custom message containing the following information:

- Activity in which the error occurred
- Date and time the error occurred
- The unique identifier of the message
- The error message generated by the system

The system administrator uses this information for tracking and debugging purposes. This error log may be viewed via a text editor or through the TIBCO Administrator GUI.

2.6.4 Upon completion of activities 2.6.1 – 2.6.3 the sub-process end and the main FeederToCFS_Outbound process resumes.

2.7 This JMS Queue Sender activity generates a JMS message and sends it to the appropriate queue in the F2C_In_Q group. The object type (Obligation or Requisition) and the receiving bureau determine the queue. TIBCO identifies the receiving bureau based on the bureau code contained in the ACCS string of the message sent from CSTARS. This will allow us to determine which bureau should receive the message even in cases where multiple bureau instances are present at one location, such as OCS which contains NOAA, OS and Census CSTARS instances. Below is the list of queues the message may be sent to:

<env>.<queue-group>.<sending-app>.<receiving-app>.<object-type>.<bureau>

- <env>.f2cinq.cstars.cfs.obligation.census
- <env>.f2cinq.cstars.cfs.requisition.census

- <env>.f2cinq.cstars.cfs.obligation.nist
- <env>.f2cinq.cstars.cfs.requisition.nist
- <env>.f2cinq.cstars.cfs.obligation.noaa
- <env>.f2cinq.cstars.cfs.requisition.noaa
- <env>.f2cinq.cstars.cfs.obligation.os
- <env>.f2cinq.cstars.cfs.requisition.os

If the JMS server is down for some reason, the process transitions to the FeederToCFS_errorlog where the appropriate error information is captured.

2.8 The FeederToCFS_errorlog creates a custom message containing the following information:

- Activity in which the error occurred
- Date and time the error occurred
- The unique identifier of the message
- The error message generated by the system

The system administrator uses this information for tracking and debugging purposes. This error log may be viewed via a text editor or through the TIBCO Administrator GUI.

2.9 The next activity writes a custom message to a log file indicating which JMS queue the message has been sent to. This message can be viewed through the TIBCO Administration GUI.

3. If the destination of the JMS message is a CFS instance at a different location (i.e. NOAA-CSTARS to NOAA-CFS) then the message must be forwarded from the JMS server of the source system to the JMS server of the receiving system. The JMS server is configured to always perform this action for queues where needed. This is done by modifying the routes.conf file of the JMS server. Messages in the F2C_In_Q group that need to be forwarded to a different location are forwarded to the F2C_In_Q group on the JMS server of the destination system.

   If a CFS or CSTARS application is migrated to a different location, the JMS server can be reconfigured to forward the message to the new location. Since each TIBCO instance at the different locations have the same JMS sender and receiver configuration, the TIBCO processes do not need to be modified to account for the relocation. Only the routing functionality of the JMS servers responsible for forwarding the messages to the appropriate location need to be reconfigured. This provides flexibility and significantly reduces the effort needed for a modification of that type.

4. The TIBCO processes at the destination location contain JMS receivers configured to listen for the incoming messages. As in step 2, a JMS receiver listens for Requisitions and a separate JMS receiver listens for Obligations. Now that the destination systems have been identified and there are separate queues for each, there are also separate processes for each possible receiving bureau. The incoming message is parsed using the appropriate XML schema definition, the necessary data mappings are executed and the message is inserted into the CFS system via JDBC calls and stored procedures. See the diagrams and sections 4.1 – 4.7 for further detail.

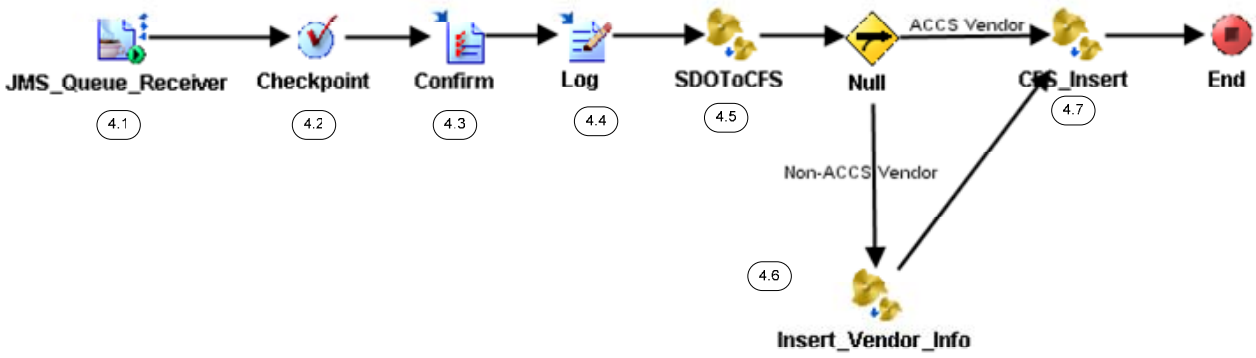


Figure 3: Feeder to CFS Inbound Process – Level 1

4.1 A configured JMS receiver listens to the F2C_In_Q grouping. There are JMS receivers configured to receive messages sent to each possible destination bureau. Each bureau destination can also receive both Obligation and Requisition messages. All together there eight separate processes accounting for each queue possibility. Four bureau possibilities, each receiving either Requisition or Obligation messages. See the F2C_In_Q grouping in the JMS configuration diagram (Figure 5). The information below lists the possible JMS queues that may be received by this process:

<env>.<queue-group>.<sending-app>.<receiving-app>.<object-type>.<bureau>

- <env>.f2cinq.cstars.cfs.obligation.census
- <env>.f2cinq.cstars.cfs.requisition.census
- <env>.f2cinq.cstars.cfs.obligation.nist
- <env>.f2cinq.cstars.cfs.requisition.nist
- <env>.f2cinq.cstars.cfs.obligation.noaa
- <env>.f2cinq.cstars.cfs.requisition.noaa
- <env>.f2cinq.cstars.cfs.obligation.os
- <env>.f2cinq.cstars.cfs.requisition.os

4.2 After the JMS message is brought into the TIBCO BusinessWorks engine a checkpoint activity is executed. A checkpoint saves the current process data and state so that it can be recovered at a later time in the event of a failure. If a process engine fails, all process instances can be recovered and resume execution at the location of their last checkpoint in the process definition.

4.3 The Confirm activity confirms any confirmable messages received by the process instance. For ORSI purposes, it confirms that the BusinessWorks process has received the JMS message. When this occurs the JMS removes the message from the queue.

4.4 The next activity writes a custom message to a log file stored within the TIBCO BusinessWorks installation directory under the \logs subdirectory. This message can be viewed through the TIBCO Administration GUI. This custom message is tailored to provide useful information which the system administrator uses for tracking and debugging purposes. This log file records the queue from which the message came as well a time stamp and the unique identifier of the message.

4.5 This activity calls the sub-process that transforms the dataset from the shared data object to the schema needed within CFS. This functionality is in a sub-process so that it may be called by other processes if the need arises. The SDO also decouples the CSTARS and CFS formats. If other entities want to leverage the enterprise integration framework in the future, only the transformation to/from the SDO needs to be developed. The diagram and steps below describe the activities within this sub-process.

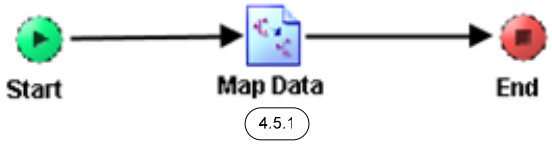


Figure 4: SDO to CFS Transformation sub-process – Level 2

4.5.1 This Map Data activity transforms information from the SDO format to the schema required for CFS processing. See Appendix A for details regarding this mapping.

4.6 If the dataset sent from CSTARS contains a non-CCR vendor, this vendor must be inserted separately into CFS by JDBC calls and stored procedures. These JDBC calls and stored procedures were created in Phase I of this standard interface development and are leveraged for use in the ORSI development.

TIBCO identifies non-CCR vendors by keying on the F2C_Vendor_Id field. If this field is null, then a CCR vendor was sent. If the field is not null then a non-CCR vendor was sent. CCR vendors do not execute this functionality and move to the next sub-process (step 4.7) to insert the Requisition or Obligation data into the CFS staging tables.

The diagram and steps below detail the functionality of this sub-process.

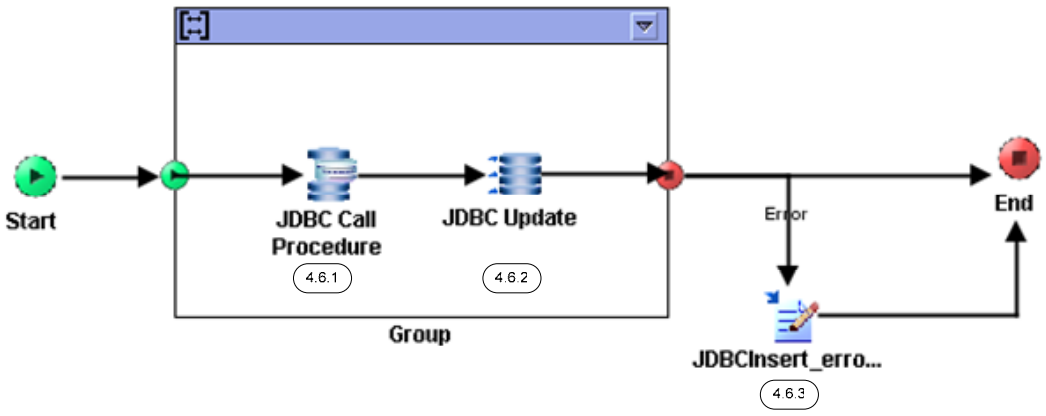


Figure 5: Feeder to CFS Vendor Insert

4.6.1 This JDBC call procedure calls a stored procedure within CFS to obtain a unique message ID that is inserted as part of the vendor dataset. This calls the Get_C2F_I_Queue_003_ID procedure within the EAI_PM002_TIBCO package of CFS. See CFS Detailed Design section for further detail.

4.6.2 The JDBC update task inserts the necessary vendor information into the C2F_I_Queue_003 and C2F_I_RQS_Vendor_Detail tables. The SQL statement from Phase I used to perform this update are leveraged in Phase II of the ORSI development.

4.6.3 The process goes to the JDBC insert error log in the event any functional or technical errors occur during the JDBC processing. This error log creates a custom message containing the following information:

- Activity in which the error occurred
- Date and time the error occurred
- The unique identifier of the message
- The error message generated by the system

This information is used by the system administrator for tracking and debugging purposes. This error log may be viewed via a text editor or through the TIBCO Administrator GUI. Please refer to section 3.4.6 for further detail regarding the Common Error Handling framework.

4.7 This sub-process updates the Obligation and Requisition staging table of CFS system. The update process is done through a series of JDBC inserts and stored procedures. The entire JDBC update process is contained in a TIBCO transaction group. A transaction group either commits or rolls back activities within the group when the transaction completes. Only JDBC activities participate in the transaction, but other activities can be part of the transaction group. If the transaction commits, all JDBC activities within the transaction group commit, if the transaction rolls back, all JDBC activities within the transaction group roll back. If an error does occur and the transaction rolls back, the process transitions to the JDBCInsert error log which is discussed in step 4.7.10.

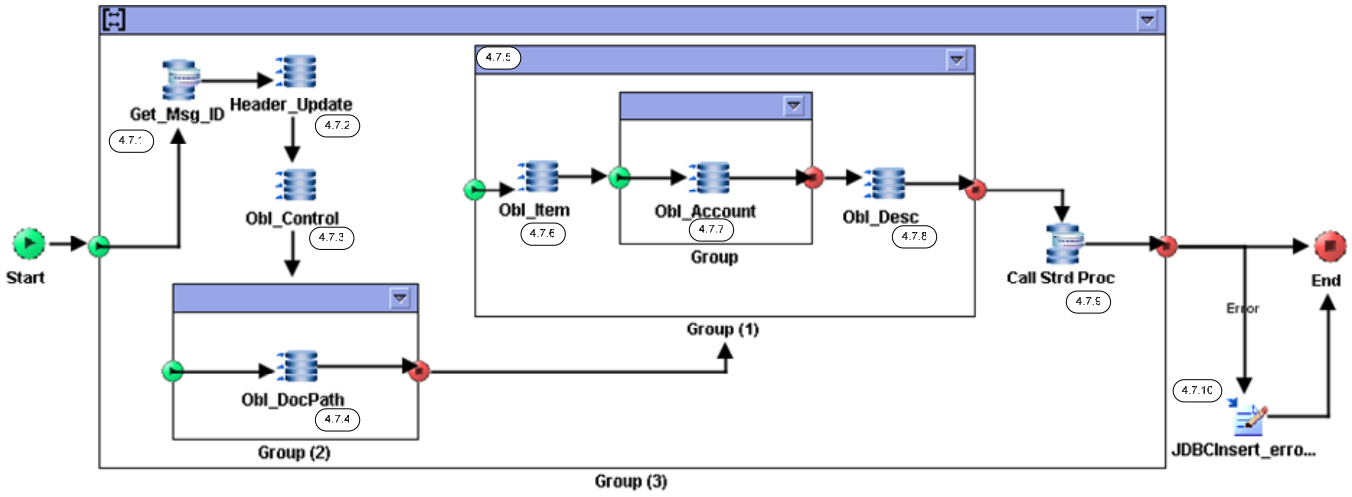


Figure 6: **Feeder to CFS Obligation/Requisition insert**

4.7.1 This JDBC call procedure calls a stored procedure within CFS to obtain a unique message ID to insert as part of the Requisition or Obligation dataset.

For Requisitions, this calls the Get_C2F_I_Queue_001_ID procedure within the EAI_FM030_TIBCO package of CFS. Please to the CFS Detailed Design section for further detail.

For Obligations, this calls the Get_C2F_I_Queue_002_ID procedure within the EAI_FM040_TIBCO package of CFS. Refer to the CFS Detailed Design section for further detail.

4.7.2 This JDBC update inserts information into the C2F_I_Queue_001 table for Requisitions and C2F_I_Queue_002 table for Obligations. SQL procedures developed in Phase I are leveraged for this task.

4.7.3 This JDBC update inserts information into the C2F_I_RQS_REQ_Control table for Requisitions and C2F_I_RQS_PO_Control table for Obligations. SQL procedures developed in Phase I are leveraged for this task.

4.7.4 The staging table updated by this activity is the C2F_I_RQS_REQ_Doc_Path staging table for Requisitions and C2F_I_RQS_PO_Doc_Path for Obligations. This update task is contained within in a TIBCO iterate task which allows looping until all records have been processed. This is necessary since there can be more than one Document Path per Requisition/Obligation.

4.7.5 Steps 4.7.6 – 4.7.8 are contained in this iteration group to allow each record to be processed. This is necessary since there may be more than one line item per transaction.

4.7.6 The C2F_I_RQS_REQ_Item and C2F_I_RQS_PO_Item staging table is updated for Requisitions and Obligations, respectively, by this JDBC update activity.

4.7.7 The C2F_I_RQS_REQ_Account staging table is updated for Requisitions and Obligations, respectively, by this JDBC update activity.

4.7.8 The C2F_I_RQS_REQ_Desc staging table is updated for Requisitions and Obligations, respectively, by this JDBC update activity.

4.7.9 This activity calls a CFS stored procedure to process a staged incoming Requisition or Obligation. The stored procedure name is Create the Process_C2F_I_Queue_001. See the CFS detailed design section for further detail regarding this stored procedure.

4.7.10 If an error occurs at any point in the process all transactions are rolled back and the process transitions to the JDBC insert error log. This error log creates a custom message containing the following information:

- Activity in which the error occurred
- Date and time the error occurred
- The unique identifier of the message
- The error message generated by the system

The system administrator uses this information for tracking and debugging purposes uses. This error log may be viewed via a text editor or through the TIBCO Administrator GUI. Please refer to section 3.4.6 for further detail regarding the Common Error Handling framework.

5. Messages inserted into the CFS response tables by CFS processing logic are received by the TIBCO Active Database (ADB) adapter services that are polling the tables. The adapter brings the messages into the TIBCO BusinessWorks processing engine. There is a process containing an ADB adapter service that polls the Requisition response tables and a separate process containing an ADB adapter service that polls the Obligation response tables. This framework uses only one Active Database adapter for each location; however multiple ADB adapter services will exist. After BusinessWorks receives the response message, it is mapped to the Shared Data Object, rendered into an XML message and sent to the appropriate JMS message queue. The diagram and steps below describe the details of this process.

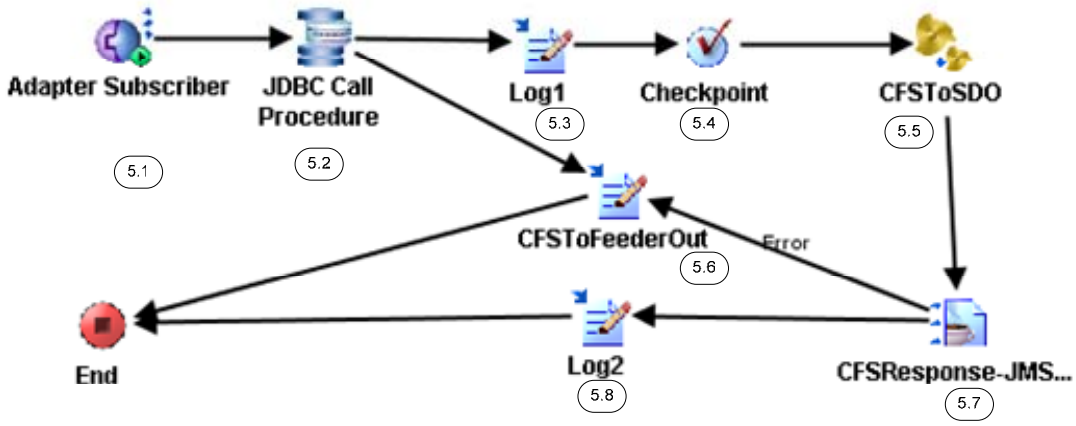


Figure 7: CFS to Feeder System Outbound Process – Level 1

5.1. This Active Database (ADB) adapter service polls the message response staging tables. When the CFS staging tables generate a new response or the status of a response is modified, the adapter service receives the message and brings it into TIBCO BusinessWorks for processing. A separate adapter service is configured for Requisition and Obligation responses. JMS adapter services are used.

5.2. A CFS stored procedure is called after the response message has been received by the BusinessWorks process. The stored procedure name is Update_C2F_O_Queue_000N and it confirms within the CFS response table that the message has been successfully accepted by the BusinessWorks process.

5.3. The next activity writes a custom message to a log file stored within the TIBCO BusinessWorks installation directory under the \logs subdirectory. This message can be viewed through the TIBCO Administration GUI. This tailored message provides useful information, which the system administrator uses for tracking and debugging purposes. This log file records the unique identifier of the response message, the receiving time and the return status.

5.4. After the message is brought into the TIBCO BusinessWorks engine a checkpoint activity is executed. A checkpoint saves the current process data and state so that it can be recovered at a later time in the event of a failure. If a process engine fails, all process instances can be recovered and resume execution at the location of their last checkpoint in the process definition.

5.5. This activity calls the sub-process that transforms the response message from CFS format to the shared data object (SDO). This functionality is in a sub-process so that it may be called by other processes if the need arises. The SDO also decouples the CSTARS and CFS formats. The diagram and steps below describe the activities within this sub-process.

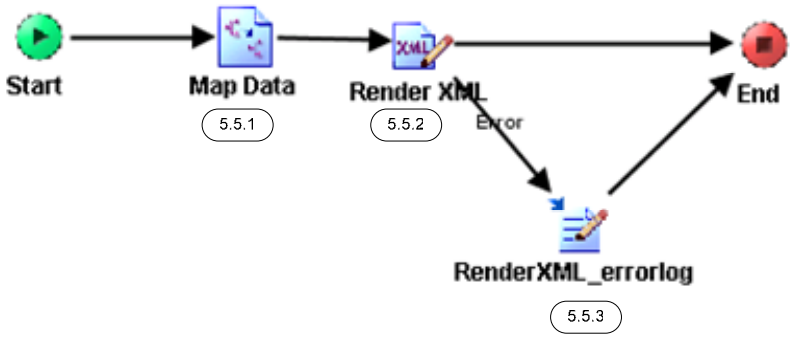


Figure 8: CFS to SDO Transformation sub-process – Level 2

5.5.1. This Map Data activity transforms information from the CFS format to the Shared Data Object schema.

5.5.2. The render XML activity creates the XML message that is forwarded within the body of a JMS message onto the appropriate queue.

5.5.3. The process transitions to the Render XML error log if it is unable to create the XML message for any reason. This error could occur due to values not being present in required elements of the XML message.

5.6. The CFS to Feeder error log creates a custom message containing the following information in the event the stored procedure in step 5.2 is unable to communicate with the database or the JMS server is down and the JMS message cannot be sent in step 5.7:

- Activity in which the error occurred
- Date and time the error occurred
- The unique identifier of the message
- The error message generated by the system

This information is used by the system administrator for tracking and debugging purposes. This error log may be viewed via a text editor or through the TIBCO Administrator GUI. Please refer to section 3.4.6 for further detail regarding the Common Error Handling framework.

5.7. This JMS Queue Sender activity generates a JMS message and sends it to the appropriate queue in the C2F_Out_Q group. The queue is determined by the object type (Obligation or Requisition) and the receiving bureau.

5.8. The next activity writes a custom message to a log file indicating which JMS queue the message has been sent to. This message can be viewed through the TIBCO Administration GUI.

6. If the destination of the JMS message is a CSTARS instance at a different location (i.e. NOAA-CFS to NOAA-CSTARS) then the message must be forwarded from the JMS server of the source system to the JMS server of the receiving system. The JMS server is configured to always perform this action for queues where needed. Messages in the C2F_Out_Q group

that need to be forwarded to a different location are forwarded to the C2F_Out_Q group on the JMS server of the destination system.

7. The response message is pulled from the C2F_Out_Q grouping by JMS receivers configured within the TIBCO BusinessWorks processes. A JMS receiver listens for Requisitions and a separate JMS receiver listens for Obligations. There are separate queues for each possible receiving bureau. The incoming message is parsed using the appropriate XML schema definition, mapped from the SDO schema to the Feeder System schema and then sent in a JMS message for the Feeder System to receive. See the diagrams and sections 7.1 – 7.5 for further detail.

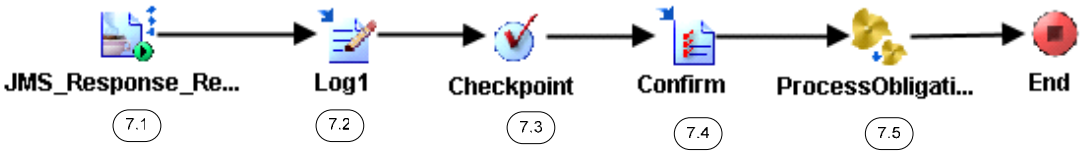


Figure 9: CFS to Feeder System Inbound Process – Level 1

7.1. A JMS receiver is configured to listen to the C2F_Out_Q grouping. There are JMS receivers configured to receive messages sent to each possible destination bureau. Each CSTARS destination can also receive both Obligation and Requisition messages. All together there are eight separate processes accounting for each queue. Four bureau possibilities, each receiving either Requisition or Obligation messages. See the C2F_Out_Q grouping in the JMS configuration diagram (figure 5).

7.2. After the JMS message is brought into the TIBCO BusinessWorks engine a checkpoint activity is executed. A checkpoint saves the current process data and state so that it can be recovered at a later time in the event of a failure. If a process engine fails, all process instances can be recovered and resume execution at the location of their last checkpoint in the process definition.

7.3. The Confirm activity confirms any confirmable messages received by the process instance. For ORSI purposes, it is used to confirm the JMS message has been received by the BusinessWorks process. When this occurs the JMS server removes the message from the queue.

7.4. The next activity writes a custom message to a log file stored within the TIBCO BusinessWorks installation directory under the \logs subdirectory. This message can be viewed through the TIBCO Administration GUI. This custom message is tailored to provide useful information which the system administrator uses for tracking and debugging purposes. This log file records the queue from which the message came as well a time stamp and the unique identifier of the message.

7.5. This activity calls a sub-process that parses the XML converts it from the SDO schema to the schema required by the Feeder System. After it has been converted the message is rendered into an XML string and sent to the appropriate queue on the JMS server. See the diagram and additional steps below for more information.

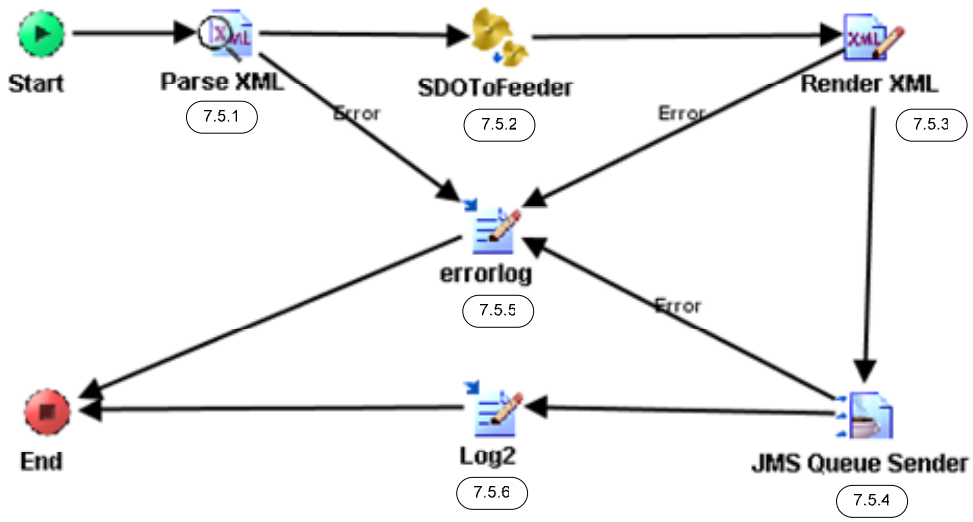


Figure 10: C2F Response Processing – Level 2

7.5.1. The XML string within the body of the JMS message is parsed using a *Parse XML* activity and the appropriate XSD. One XSD is created for Requisitions and one for Obligations since each contains different fields. Each is used in its respective process. This activity separates the raw XML string into name/value pairs for translation, interpretation and mapping.

7.5.2. This activity calls the sub-process that transforms the response message from Shared Data Object (SDO) to the Feeder System Schema. This functionality is in a sub-process so that it may be called by other processes if the need arises. The diagram and steps below describe the activities within this sub-process.

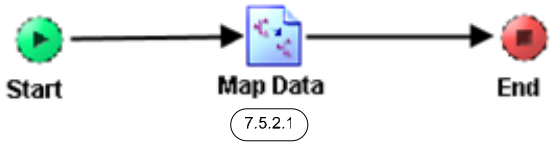


Figure 11: SDO to Feeder System Transformation sub-process – Level 3

7.5.2.1. This Map Data activity transforms information from the SDO format to the schema required for Feeder System processing.

7.5.3. The render XML activity creates the XML message that is forwarded within the body of a JMS message onto the appropriate queue.

7.5.4. This JMS Queue Sender activity generates a JMS message and sends it to the appropriate queue in the C2F_In_Q group. The queue is determined by the object type (Obligation or Requisition) and the CSTARS bureau instance.

7.5.5. If an error occurs in the parsing or rendering of the XML message or if JMS communication is unavailable, the process transitions to the error log of the process. The error log captures the following standard information:

- Activity in which the error occurred
- Date and time the error occurred
- The unique identifier of the message
- The error message generated by the system

This information is used by the system administrator for tracking and debugging purposes. This error log may be viewed via a text editor or through the TIBCO Administrator GUI.

7.6. A custom message is written to a log file indicating which JMS queue the message has been sent to. This message can be viewed through the TIBCO Administration GUI.

8. Messages sent to the C2F_In_Q group are consumed by the JMS receivers configured on the Feeder System application.